Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 29 (2012) 1281 – 1285

Procedia Engineering

www.elsevier.com/locate/procedia

2012 International Workshop on Information and Electronics Engineering (IWIEE)

MSDL based Lifecycle Support of Cloud Simulation

Cai Ying[a]*, Peng Yong[a], Huang Kedi[a], Huang Qingyun[b]

[a]*College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha, Hunan, P.R.China*
[b]*Operational Command Teaching & Research Section, Air Defense Academy, 24 East Jianshe Rd.,Zhenzhou, Henan, P.R. China.*

Abstract

To realize cloud simulation, automation of process is the key technology that should be settled. Based on the analysis of the characteristic of simulation, a MSDL (Military Scenario Definition Language) based framework is proposed to provide the lifecycle support of cloud simulation. MSDL based resource discovery and schedule mechanisms are also presented. Analysis shows that MSDL based framework can help automatically fulfill simulation in cloud computing environment.

Keywords: Cloud simulation, Automation of process, MSDL, Cloud computing

1. Introduction

The rapid development of network technologies is changing the way how the engineers in modeling and simulation (M&S) fields think and design. Thanks to the internet more resources are shared and further cooperation can be undertaken. Even more exciting is the recent development of cloud computing which provides a compelling value proposition for organizations to outsource and obtain resources on demand over the Internet[1][2], which also can be used for reference in simulation [3][4]. Cloud computing is supposed to transparently fulfill the needs submitted by clients, which can be favorable for all engineers in M&S. However, before it is applied to simulation, several problems should be settled during the lifecycle of a simulation, especially:

* Corresponding author. Tel.: 00086+15973121396.
E-mail address: caiying_cn@126.com..

doi:10.1016/j.proeng.2012.01.127

Firstly, how to choose the right resources? Since the problem for M&S has changed from resource shortage to resource combination explosion, it is unrealistic for the customers to choose the resources manually especially considering the huge number of resources needed for a military operation simulation.

Secondly, how to schedule the services in the lifecycle of a simulation? The lifecycle of simulation involves many steps; each deals with complex operations and collaborations. Cloud simulation should provide a mechanism for the automation of the whole process.

As an important element in M&S, Scenario is like a bridge between concept model and execution of a simulation, it is both an embodiment of concept model and an abstract for the execution of a simulation. Scenario includes the models, services, tasks, COA (course of action) and the context for running a simulation, which can be of great help for the automation design of cloud simulation. In this paper we mainly focus on military domain. Standards are studied and MSDL based mechanisms are proposed to support the lifecycle of simulation in cloud computing environment.

2. Scenario in MSDL

Military scenario is a specific description of the situation and course of action at a moment in time for each element in the scenario. Whether the content in scenario is understandable can greatly influent the interoperability. A standard should be made so that scenarios compliant to it can interoperate with each other. That's why SISO raised and approved MSDL as a standard in 2008 to provide the mechanism that permits simulations to utilize the MSDL schema to develop and reuse military scenarios across MSDL compliant simulations and scenario generation tools.

2.1. Brief introduction of MSDL

The Military Scenario Definition Language (MSDL) is an XML-based language designed to support military scenario development that provides the modelling and simulation community with [5]:

- A common mechanism for verifying and loading military scenarios;
- The ability to create a military scenario that can be shared between simulations and C4I devices;
- A way to improve scenario consistency between federated simulations;
- The ability to reuse military scenarios as scenario descriptions are standardized throughout the Army, Joint, and international communities and across simulation domains, e.g. training exercise, analysis, etc.

As is shown in Fig. 1, the current version of MSDL specification mainly includes 9 components, namely: ScenarioID, Options, Environment, ForceSides, Organizations, Overlays, Installations, TacticalGraphics, and MOOTWGraphics.

MSDL is not only used within the planning phase of simulation for breaking and fuse scenarios but also within the execution phase to enable interoperability of simulation and operations [6].

However, current version of the MSDL standard in balloting under SISO does not contain a specification of the COA structure. It is intended that this structure will be defined by the C-BML specification when that becomes available and approved [7].

3. MSDL based simulation design

The simulation experiment can be divided into three phases: preparation phase, execution phase and result treatment phase, as is shown in shadow frame in Fig 2. In the preparation period, central work is to find right resources to meet the needs defined in scenario. When it comes to execution phase, central work turns to scheduling. Sometimes resource discovery on the fly is needed when scenario changes. In the end,

results are treated for estimation and display, etc. All these seeming irrelevant operations are in effect related to the scenario. Therefore by using MSDL-a standard scenario schema, simulation can be carried out in an automated manner and the framework is shown in Fig. 2.

Fig. 1.Top-Level MSDL Schema Structures [5]

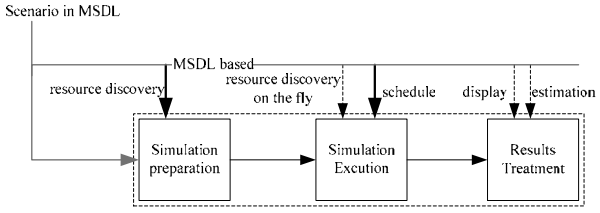


Fig. 2 MSDL based lifecycle support framework

Since resource discovery and process scheduling are two compulsory jobs, they're detailed in the following.

3.1. MSDL based resource discovery

3.1.1 Problem statement

In cloud simulation environment, when a customer requests for resources, the request is dealt with by the discovery agent, the discovery agent compares the request with registered information and returns the one that matches the most. Three elements are necessary for this process:

- the description of the resources;
- the representation of request;
- the discovery model and matching algorithms

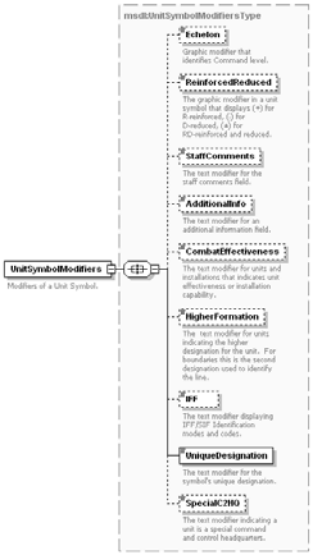
The first and third elements have been widely studied in the web service discovery which can be used for reference. However, the second element is often neglected mainly because is considered to be the customer's work, which may cause problems because to combine sufficient and understandable information in the request is the base for the whole discovering process. On the other hand, to represent the information is very hard work considering the numbers of model needed in one simulation.

3.1.2 Problem solved ways

In MSDL, model resource requirements are defined in Unit element and Equipment element in Organizations component. Thus there is no need to do extra work to make a request. Since MSDL has adopted many standard schemas and data types, it's easy to interoperate between the discovery models..

The UnitSymbolModifiers element in Unit element specifies the modifiers of unit symbol. It is an xs:all compositor comprised of the elements of the elements shown in Fig. 3, including almost all the information needed for discovering the model.

We designed the discovery model as is shown in Fig. 4.



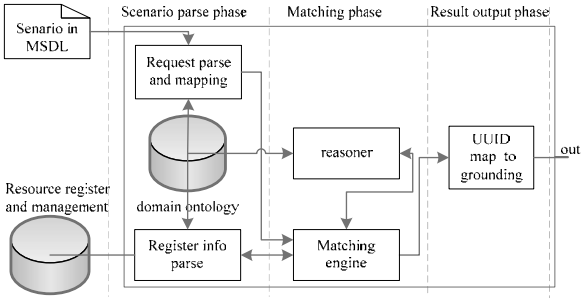


Fig. 3 UnitSymbolModifiers Element Structure [5]

Fig. 4 Simulation resource discovery model

For every matching process, there are three steps. Firstly the scenario is parsed by discovery agent and the request for the simulation is divided to single requests for each model; secondly, the match engine compares the requests with the registered resources, makes reasoning based on domain knowledge base, calculates the similarity and returns the top scored result; at last, the result is mapped to the universal unique identifier of Unit.

Since the models can be searched separately, it can be conducted in parallel. Descriptions of model in UnitSymbolModifiers Element Structure are converted to relation diagram which makes the querying process more efficient.

3.2. MSDL based control mechanism

Collaborated simulation can be difficult because the time advance relations between federates often are complex. Traditionally the engineer should manually turn the scenario into a running script for the computer to follow. It wastes a lot of time and the running script can be hard for reuse due to heterogeneity. The COA element of a scenario can reflect this information. However current version of MSDL does not have this element specified due to certain reasons [7]. We take this information in the COA element from an earlier version [9]. The “5Ws” – “Who”, “What”, “When”, “Where”, and “Why” are defined as is shown in Fig. 5. The units (through UnitOwnerHandle element to find) represented within MSDL file provide the Who. The MSDL Task Element provides the “What”. The Trigger data (through TriggerHandle element to find) provides the “When” and “Where” the task will be triggered. The Why is provided as part of the Task Element.

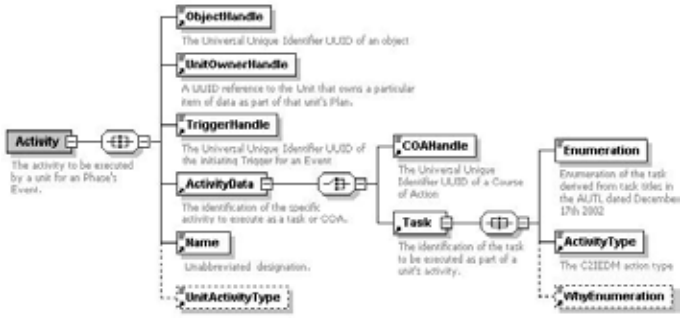


Fig.5 “5W” in MSDL

Based on this model, it is simple to control the process according to the COA defined in a MSDL style scenario. Since MSDL is standardized, control system can easily parse this information and execute automatically.

4. Conclusions

Using MSDL to define Scenario not only promotes interoperability between different organizations but also between different phases of simulation. With proper design, MSDL can support the lifecycle automation of cloud simulation in the following ways:

- MSDL supports automated resource discovery before and during the execution of simulation;
- MSDL supports automated schedule of simulation execution;

References

[1] Ian Foster, Cloud Computing and Grid Computing 360-Degree Compared , GCE2008, 2008:1-10.

[2] Paul Watson, Cloud Computing for e-Science with CARMEN, PDCAT2007, 2007(5): 5.

[3] Bohu Li, Xudong Cai, et al., Networked Modeling and Simulation Platform based on the Concept of Cloud Computing-Cloud Simulation, Journal of System Simulation, 2009, 21(7):5292-5299

[4] Cai Ying, Yang Mei, Huang Kedi. Resource Sharing Architecture for Simulation Based on Hybrid pattern . *Journal of System Simulation*, 2011,7

[5] Simulation Interoperability Standards Organization Military Scenario Definition Language Study Group (SG). Standard for: Military Scenario Definition Language (MSDL), SISO-STD-007-2008: *http://www.sisostds.org/digitalLibrary.aspx?command=core_Download&EntryId-29782* last access June 10, 2011.

[6] R. W. Franceschini, D. J. Franceschini, et al., Specifying Scenarios Using the Military Scenario Definition Language, *2004 fall SIW,* paper 04F-SIW-048

[7] Curtis Blais, Military Scenario Definition Language (MSDL): How Broadly Can It Be Applied, *2008 Spring SIW,* 08S-SIW-002

[8] Simulation Interoperability Standards Organization Military Scenario Definition Language Study Group (SG), Specifications for: Military Scenario Definition Language (MSDL), April 2005.

[9] Saikou Diallo, Dr. Andreas Tolk, et al., An Application Extension for the Military Scenario Description Languag, *2006 Spring SIW,* 06S-SIW-009